

# Adjoint Formulation for an Embedded-Boundary Cartesian Method

Marian Nemec,\* Michael J. Aftosmis,<sup>†</sup> Scott M. Murman,<sup>‡</sup>  
and Thomas H. Pulliam<sup>§</sup>

*NASA Ames Research Center  
MS T27B, Moffett Field, CA 94035*

Abstract submitted to the 43rd AIAA Aerospace Sciences Meeting  
Jan. 10–13, 2005/Reno, NV

## I. Introduction

MANY problems in aerodynamic design can be characterized by smooth and convex objective functions. This motivates the use of gradient-based algorithms, particularly for problems with a large number of design variables, to efficiently determine optimal shapes and configurations that maximize aerodynamic performance. Accurate and efficient computation of the gradient, however, remains a challenging task. In optimization problems where the number of design variables dominates the number of objectives and flow-dependent constraints, the cost of gradient computations can be significantly reduced by the use of the adjoint method.

The problem of aerodynamic optimization using the adjoint method has been analyzed and validated for both structured and unstructured grids. The method has been applied to design problems governed by the potential, Euler, and Navier–Stokes equations and can be subdivided into the continuous<sup>1,2,3,4</sup> and discrete formulations.<sup>5,6,7,8,9,10,11</sup> Giles and Pierce<sup>12</sup> provide a detailed review of both approaches. Most implementations rely on grid-perturbation or mapping procedures during the gradient computation that explicitly couple changes in the surface shape to the volume grid. The solution of the adjoint equation is usually accomplished using the same scheme that solves the governing flow equations. Examples of such code reuse include multistage Runge–Kutta schemes coupled with multigrid,<sup>3,9</sup> approximate-factorization,<sup>6</sup> line-implicit Gauss–Seidel,<sup>8,13</sup> and also preconditioned GMRES.<sup>10,11</sup>

The development of the adjoint method for aerodynamic optimization problems on Cartesian grids has been limited. In contrast to implementations on structured and unstructured grids, Cartesian grid methods decouple the surface discretization from the volume grid.<sup>14</sup> This feature makes Cartesian methods well suited for the automated analysis of complex geometry problems, and consequently a promising approach to aerodynamic optimization. Melvin *et al.*<sup>15</sup> developed an adjoint formulation for the TRANAIR code,<sup>16</sup> which is based on the full-potential equation with viscous corrections. More recently, Dadone and Grossman<sup>17</sup> presented an adjoint formulation for the Euler equations. In both approaches, a boundary condition is introduced to approximate the effects of the evolving surface shape that results in accurate gradient computation.

---

\*NRC Research Associate, Member AIAA

<sup>†</sup>Research Scientist, Member AIAA

<sup>‡</sup>ELORET Senior Research Scientist, Member AIAA

<sup>§</sup>Senior Research Scientist, Associate Fellow AIAA

In Ref. 18, we presented an effective framework for complex geometry aerodynamic optimization problems. Key modules of the framework were a Cartesian inviscid-flow analysis package, *Cart3D*,<sup>19,20</sup> and a direct-CAD interface.<sup>21</sup> A genetic and gradient-based algorithms were used to drive the optimization procedure. The gradients were computed via finite-differences.

In this work, we improve the efficiency of the gradient computation by developing an adjoint method for aerodynamic optimization on Cartesian grids with embedded boundaries. We present the numerical implementation for the discrete approach, where the adjoint solver efficiently reuses the domain decomposition, multigrid, and time-marching schemes of the flow solver. We focus on the accuracy of the gradient computation, which requires careful treatment of cut-cells, the underlying surface triangulation, and the interface to a parametric-CAD system. *The final paper will contain a number of complex geometry, industrially relevant, examples with many design variables to demonstrate the effectiveness of the adjoint method on Cartesian grids.*

## II. Problem Formulation

The aerodynamic shape optimization problem consists of determining values of design variables  $X$ , such that the objective function  $\mathcal{J}$  is minimized

$$\min_X \mathcal{J}(X, \bar{Q}) \quad (1)$$

subject to constraint equations  $C_j$

$$C_j(X, \bar{Q}) \leq 0 \quad j = 1, \dots, N_c \quad (2)$$

where the vector  $\bar{Q}$  denotes the continuous, conservative flow variables and  $N_c$  denotes the number of constraint equations. The flow variables are forced to satisfy the governing flow equations within a feasible region of the design space  $\Omega$

$$\mathcal{F}(X, \bar{Q}) = 0 \quad \forall X \in \Omega \quad (3)$$

which implicitly defines  $\bar{Q} = f(X)$ .

The objective function defines the goals of the optimization problem, while the constraint equations limit the feasible region of the design space. The constraints may involve performance functionals, such as lift, geometric quantities, such as volumes and thicknesses, and also simple bound constraints for design variables. Performance objectives can be specified for the entire configuration or for a specific subset of components, for example moments on control surfaces. This is accomplished by using the Geometry Manipulation Protocol,<sup>22</sup> where we specify component hierarchies for parametric-CAD assemblies and triangulations to intuitively reflect aerodynamic design goals.

A modular framework<sup>18</sup> is used to solve the optimization problem defined by Eqs. 1–3. We cast the optimization problem as an unconstrained problem by lifting the side constraints, Eq. 2, into the objective function using a penalty method. The constraint imposed by the flowfield equations, Eq. 3, is satisfied at every point within the feasible design space, and consequently these equations do not explicitly appear in the formulation of the optimization problem. An unconstrained BFGS quasi-Newton algorithm coupled with a backtracking line search<sup>23,24</sup> is used to find the optimal solution. At the core of the optimization framework is the analysis module, which consists of a CAD system interface provided by CAPRI,<sup>25,21</sup> a Cartesian grid

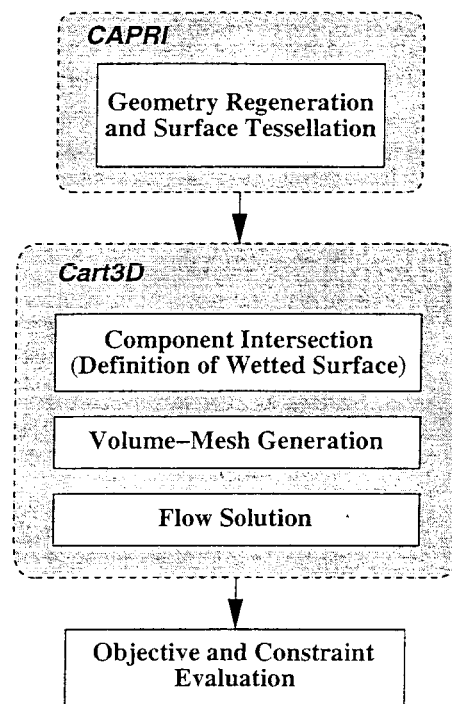


Figure 1. Components of the analysis module

generator for component-based geometry,<sup>19</sup> and a flow solver, as outlined in Fig. 1. Below, we provide a brief description of the flow solver. Thereafter, we focus on the development of the gradient computation algorithm.

### III. Flow Solver

The governing flow equations are the three-dimensional steady-state Euler equations of a perfect gas. A second-order accurate finite-volume spatial discretization is used, which is based on van Leer's flux vector splitting.<sup>20</sup> The resulting system of nonlinear equations is given by

$$R(Q, X) = 0 \quad (4)$$

where  $Q = [\rho, \rho u, \rho v, \rho w, \rho E]^T$  denotes the discrete vector of integral cell averages. In each cell of the volume grid, the residual vector can be expressed as

$$R_i = \frac{1}{V_\Omega} \sum_{\partial\Omega} F \cdot \hat{n} S \quad (5)$$

where  $V_\Omega$  is the cell volume,  $\partial\Omega$  are the faces of the cell,  $F$  is the flux vector, and  $\hat{n}$  is the face unit normal. Although  $R$  is written as a function of the design variables, we emphasize that during a flow solution the design variables remain constant. A solution is achieved by introducing an unsteady term given by

$$\frac{dQ}{dt} + R(Q) = 0 \quad (6)$$

and marching to steady-state using a Runge-Kutta scheme in conjunction with a parallel multigrid method.<sup>20</sup>

### IV. Adjoints and Sensitivities

The gradient,  $G$ , of the objective function  $\mathcal{J}[X, Q(X)]$  is given by

$$G = \frac{d\mathcal{J}}{dX} = \frac{\partial\mathcal{J}}{\partial X} + \frac{\partial\mathcal{J}}{\partial Q} \frac{dQ}{dX} \quad (7)$$

where  $Q$  denotes a discrete steady-state solution of the flow equations. We reduce the vector of design variables,  $X$ , to a scalar in order to clearly distinguish between partial and total derivatives. For problems with multiple design variables, it may be helpful to note that  $G$  and  $\partial\mathcal{J}/\partial X$  are  $[1 \times N_D]$  row vectors,  $\partial\mathcal{J}/\partial Q$  is a  $[1 \times N_F]$  row vector, and  $dQ/dX$  is a  $[N_F \times N_D]$  matrix, where  $N_D$  and  $N_F$  represent the number of design and flow variables, respectively.

Throughout this development, we assume that the implicit function  $Q(X)$  is sufficiently smooth. Theoretically, this assumption is violated at flow discontinuities, for example shock waves, and leads to the requirement for additional boundary conditions. In practice, the smoothing of the discontinuities by the underlying spatial discretization of the flow equations provides a  $Q(X)$  that is sufficiently smooth.<sup>26,12</sup>

In Eq. 7, the evaluation of the term  $dQ/dX$ , referred to as the flow sensitivities, is obtained by differentiating Eq. 4 with respect to the design variables

$$\frac{dR}{dX} = \frac{\partial R}{\partial X} + \frac{\partial R}{\partial Q} \frac{dQ}{dX} \quad (8)$$

Realizing that  $\frac{dR}{dX} = 0$ , since for any design variable Eq. 4 is always satisfied, Eq. 8 simplifies to the following large sparse system of linear equations

$$\frac{\partial R}{\partial Q} \frac{dQ}{dX} = -\frac{\partial R}{\partial X} \quad (9)$$

The direct, or flow-sensitivity, method results from solving Eq. 9 for the flow sensitivities  $dQ/dX$  and using these values in Eq. 7 to obtain the gradient.

In order to formulate the discrete-adjoint method, substitute Eq. 9 into Eq. 7 to obtain

$$\frac{dJ}{dX} = \frac{\partial J}{\partial X} - \frac{\partial J}{\partial Q} \left( \frac{\partial R}{\partial Q} \right)^{-1} \frac{\partial R}{\partial X} \quad (10)$$

From the triple-product term in Eq. 10, define the following intermediate problem

$$\psi^T = \frac{\partial J}{\partial Q} \left( \frac{\partial R}{\partial Q} \right)^{-1} \quad (11)$$

where  $\psi$  is a  $[N_F \times 1]$  column vector. Post-multiplication of both sides by  $\partial R/\partial Q$  and applying the transpose operator results in the following linear system of equations

$$\frac{\partial R}{\partial Q}^T \psi = \frac{\partial J}{\partial Q}^T \quad (12)$$

This is known as the adjoint equation, and the vector  $\psi$  represents the adjoint variables. Substituting  $\psi$  into Eq. 10, the expression for the gradient becomes

$$\frac{dJ}{dX} = \frac{\partial J}{\partial X} - \psi^T \frac{\partial R}{\partial X} \quad (13)$$

Note that Eq. 9 is a linear system with multiple right-hand sides dependent on the number of design variables. In contrast, Eq. 12 is independent of the design variables, which is the reason for the efficiency of the adjoint method.

## A. Numerical Method

The parallel multigrid method from the flow solver is adopted to solve both the adjoint and flow-sensitivity equations. We consider the flow sensitivity equation first, since with this approach the explicit linearization of the residual equations is not necessary. The matrix-vector product on the left-hand side of Eq. 9 is approximated as follows,

$$\frac{\partial R}{\partial Q} v = \frac{R(Q + \epsilon v) - R(Q)}{\epsilon} \quad (14)$$

where  $v = dQ/dX$  and  $\epsilon = \sqrt{(\epsilon_m)/\|v\|}$ . For the cases considered in this work, we set  $\epsilon = 1 \times 10^{-4}$  in order to simplify the numerical implementation of Eq. 14. To solve for the flow-sensitivities, let

$$Q^* = Q + \epsilon v \quad (15)$$

and introduce an unsteady term in Eq. 9 to obtain the following expression

$$\frac{dQ^*}{dt} + R(Q^*) = R(Q) - \epsilon \frac{\partial R}{\partial X} \quad (16)$$

The only difference between Eqs. 16 and 6 is an extra right-hand-side source term. Recall that  $Q$  is held fixed and we set the initial vector  $v$  to zero. Hence, the solution algorithm used for the flow equations can be recycled to solve the flow-sensitivity equation with only a minor modification to the multigrid forcing function. We expect the same asymptotic convergence rate for both algorithms. A limitation of this approach is the choice of the constant stepsize  $\epsilon$ . In our experience, the approach works well for subsonic flows and serves as a useful debugging tool for establishing the accuracy of the gradient computation. The final paper will include a detailed discussion of the adjoint solver.

The partial derivative term  $\partial\mathcal{J}/\partial Q$  in Eqs. 7 and 12 is evaluated using finite differences. Note that care must be taken to ensure that the evaluation of the objective function is consistent with the enforcement of the boundary conditions in the residual equations. Otherwise, the adjoint solution may contain oscillations near wall boundaries, as demonstrated by Lu and Darmofal.<sup>27</sup> We plan to investigate this further in the final paper.

The remaining partial derivative terms in Eqs. 7 and 13, namely the objective function sensitivity  $\partial\mathcal{J}/\partial X$  and the residual sensitivity  $\partial R/\partial X$ , are usually approximated with finite differences as well. Although in some instances the partial derivatives can be obtained analytically, finite differences are necessary when a CAD system is used to parameterize the geometry. For example, the centered-difference formula for the residual sensitivities is given by

$$\frac{\partial R}{\partial X_n} = \frac{R(X + he_n, Q) - R(X - he_n, Q)}{2h} \quad (17)$$

where  $e_n$  denotes the  $n$ th unit vector,

$$h = \max(\epsilon_0 \cdot |X_n|, \sqrt{\epsilon_m}) \quad (18)$$

and  $n = 1, \dots, N_D$ . Typical value of  $\epsilon_0$  is  $1 \times 10^{-3}$ . It is important to realize that Eq. 17 involves two evaluations of only the residual vector per design variable and *not* two flow solutions, i.e.  $Q$  is constant.

Before presenting our numerical implementation for the computation of  $\partial\mathcal{J}/\partial X$  and  $\partial R/\partial X$  on Cartesian grids, it is insightful to briefly review the implementations on structured and unstructured grids. The standard finite-difference approach relies on a grid-perturbation strategy to smoothly “convect” the nodes of the volume grid with the perturbed surface geometry. This results in an accurate and efficient evaluation of the partial derivative terms. Alonso *et al.*<sup>28</sup> have successfully extended this approach to a CAD-based design environment by introducing geometry patches within the parametric-CAD model.

An important observation is that the grid-perturbation strategy explicitly couples shape sensitivities to volume grid sensitivities. However, a grid-perturbation without a corresponding surface perturbation should not, in principle, influence the objective function gradient. On the basis of this argument, Jameson and Kim<sup>29</sup> introduced a reduced adjoint gradient formulation where only surface perturbations are considered. Their results indicate that for shape optimization problems the reduced approach works well. Similar results are also presented by Soto and Löhner.<sup>30</sup> In contrast, work by Anderson and Venkatakrishnan<sup>4</sup> and later by Nielsen and Anderson<sup>31</sup> shows that for cases that involve rigid body motion, for example flap position optimization, the use of a grid-perturbation strategy is important. This appears to be related to the treatment of the trailing-edge singularity.

Turning our attention back to embedded-boundary Cartesian grids, we note that a perturbation of the surface shape affects only a few near-by cells of the volume grid. Therefore, the extent of grid sensitivities is naturally limited to roughly  $\mathcal{O}(N^2)$  cells for a volume grid with  $\mathcal{O}(N^3)$  cells. However, a straightforward implementation of finite differences is complicated by the emergence and disappearance of cut-cells at the wall boundary, as well as the relative motion of the discretized surface with respect to the volume grid.

To circumvent these difficulties, we use the following approach. For  $\partial\mathcal{J}/\partial X$ , the converged flow solution is reconstructed to the vertices of the underlying surface triangulation. Given a small perturbation, a new surface triangulation is generated using the CAPRI interface. We force the new triangulation to have a one-to-one triangle mapping with respect to the baseline triangulation. This is accomplished by using the parametric values of the baseline triangulation when tessellating the perturbed surface. The objective function values required for the finite-difference approximation are now evaluated using the triangulations instead of the volume grid. Hence, the potentially non-smooth changes in the cut-cells are avoided, and the enforcement of constant  $Q$  in the evaluation of this partial derivative term becomes a trivial task. This approach works well for both shape and rigid-body motion design variables.

For the remaining term  $\partial R/\partial X$ , we have implemented and studied two approaches, both based on finite-differences. In the first approach, we assume that the only parameter dependent on the design variables

in Eq. 4 is the surface normal. This approach is similar to the “transpiration” boundary condition used in TRANAIR.<sup>32,15</sup> It is relatively easy to implement, and the evaluation of  $\partial R/\partial X$  is fast since the volume grid remains unchanged. The accuracy of this approximation can be sufficient when the design variables involve shape changes normal to the surface including wing-twist distributions. However, a possible limitation of this approach may be rigid-body motion.

In the second approach, we construct a new volume grid for the perturbed surface triangulation. The baseline solution is interpolated to the new grid via a fast interpolation algorithm presented in Ref. 33. We evaluate the residual on the new grid, which is then passed back to the baseline grid for the finite-difference computation. Cut-cells that appear or disappear during this procedure are tagged, and we either zero their contribution to the residual sensitivity computation, or we interpolate a value from their face neighbors. We evaluate both approaches in the following section.

## V. Results and Discussion

In this abstract we study a simple, two-dimensional design example to validate our approach. The baseline geometry is the NACA 0012 airfoil parameterized with a cubic B-spline curve using 17 control points. We choose the vertical motion of a control point near the leading edge of the airfoil as the design variable, shown in Fig. 2. The freestream Mach number is 0.5 and the angle of incidence is fixed at 2 deg. The goal of the optimization is lift enhancement. The baseline  $C_L$  is 0.245 and we specify a target  $C_L$  of 0.5. The “volume” grid contains roughly 9,000 cells, but since this is a two-dimensional problem, the depth in the span-wise direction contains only two cells. To help validate the results, we solve a similar problem using *Optima2D*,<sup>11</sup> which is an aerodynamic optimization tool based on a structured-grid approach.

We begin by examining the computation of the  $\partial J/\partial X$  term in Eq. 7. We evaluate the partial derivative of the lift functional (objective function) for the above problem with the angle of incidence set to 0 and 2 deg. Note that at 0 deg., this partial derivative term should vanish, providing a good validation exercise. The result at 0 deg. is indeed zero, and at 2 deg. we obtain  $-2.81 \times 10^{-3}$ . *Optima2D* predicts  $-1.4 \times 10^{-2}$ , which is a reasonable agreement considering the differences of the two approaches.

Prior to solving the flow-sensitivity equation, we evaluate the residual sensitivity  $\partial R/\partial X$ . Figure 3(a) shows  $\partial R/\partial X$  values for the y-momentum equation. These are obtained by constructing (or re-cutting) a new volume grid for the perturbed shape and interpolating the steady-state solution. Figure 3(b) shows the results when only the surface normals are changed. The agreement in Fig.3 is excellent; however, we note that for the remaining field variables significant errors may arise due to this approximation. We plan to investigate this further in the final paper.

The solution of the flow-sensitivity equation is shown in Fig. 4. Note that the zone of influence for the shape design variable extends from the leading edge to roughly 60% chord on the upper surface. The convergence of the flow and sensitivity equations is shown in Fig. 5. The flow equations converge in 500 multigrid cycles (only a 2-level multigrid is used), and the flow sensitivities require additional 400 cycles. The final gradient value is  $-1.35$ . *Optima2D* computes a value of  $-1.67$ , while a finite-difference estimation of the gradient by recomputing the flow gives  $-1.55$ . A grid-refinement study will be included in the final paper to demonstrate gradient accuracy. Furthermore, the final paper will contain additional design examples to demonstrate the effectiveness of the proposed approach.

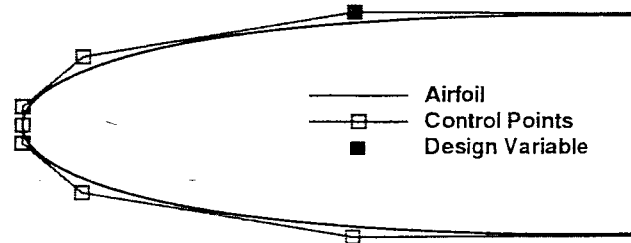
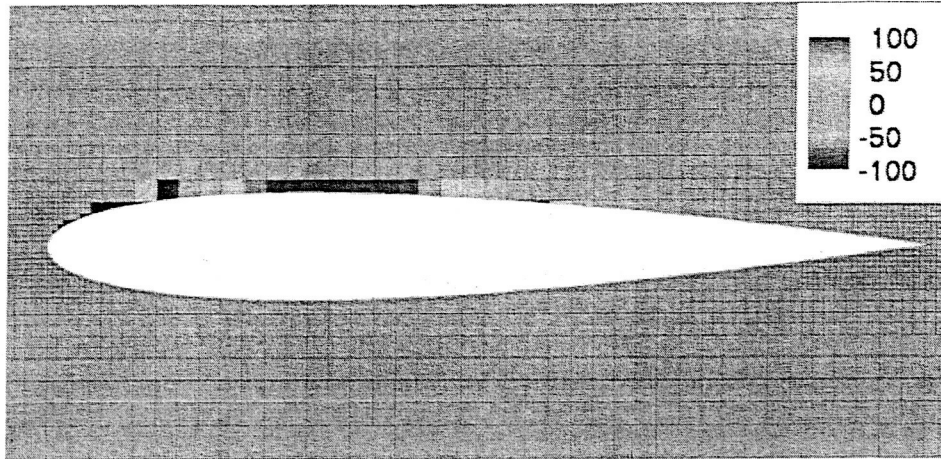
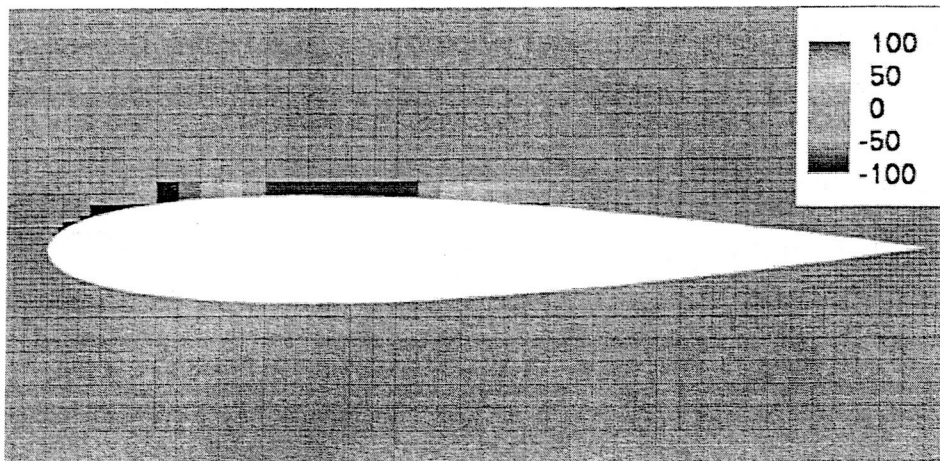


Figure 2. Design variable and B-spline parameterization of the NACA 0012 airfoil

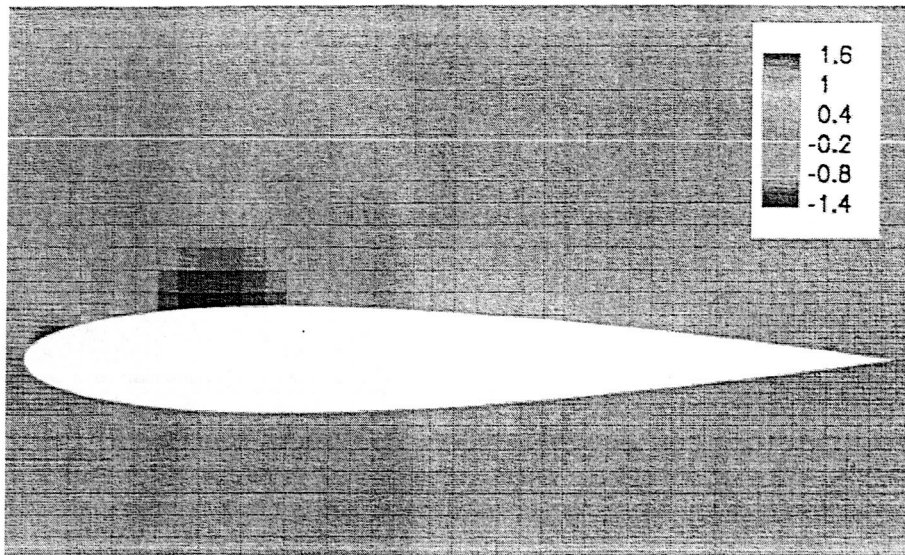


(a) Sensitivity of Eq. 4 via the grid re-cutting and interpolation approach

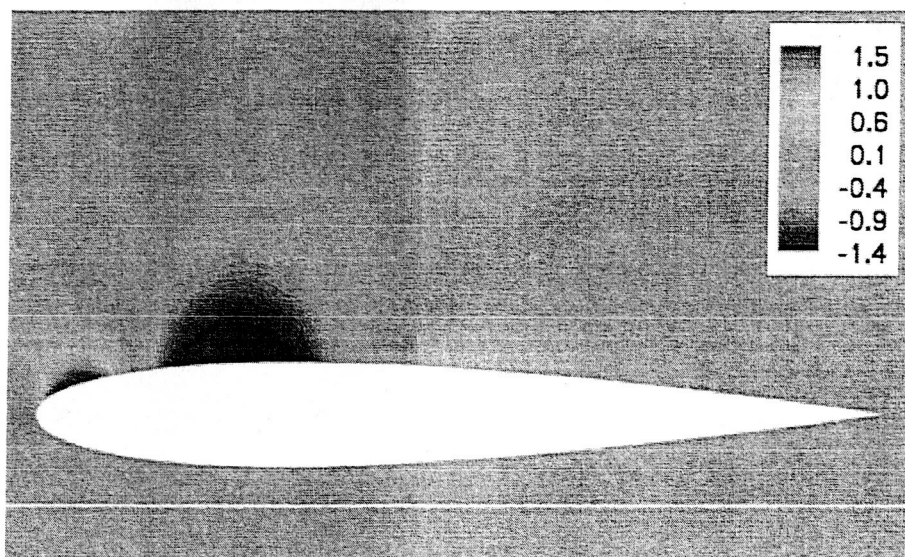


(b) Sensitivity of Eq. 4 using variations of only surface normals

Figure 3. Contours of  $\partial R/\partial X$  for the y-momentum equation



(a) Cartesian-grid formulation



(b) Structured-grid approach using *Optima2D*<sup>11</sup>

Figure 4. Contour plots of flow sensitivity: changes in density with respect to a shape design variable



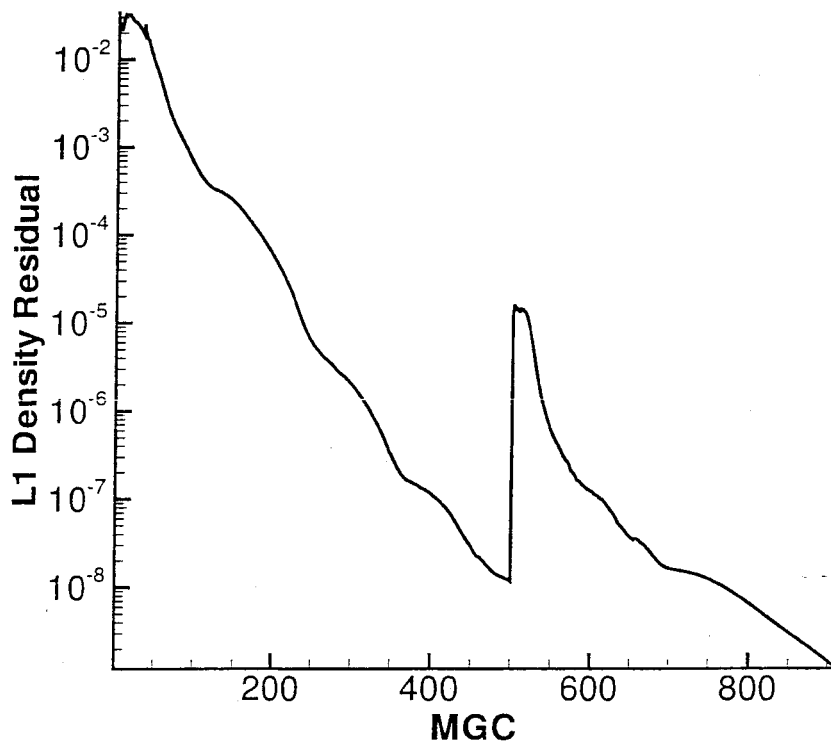


Figure 5. Convergence to steady-state for the mean flow and the flow-sensitivity equation

## VI. Acknowledgments

This work was performed while the first author held a National Research Council Research Associateship Award at the NASA Ames Research Center.

## References

- <sup>1</sup>Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260, Also ICASE report 88–64.
- <sup>2</sup>Reuther, J. J., *Aerodynamic Shape Optimization Using Control Theory*, Ph.D. thesis, University of California Davis, 1996.
- <sup>3</sup>Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamic Design using the Navier-Stokes Equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1, 1998, pp. 213–237.
- <sup>4</sup>Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *Computers & Fluids*, Vol. 28, 1999, pp. 443–480.
- <sup>5</sup>Baysal, O. and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, 1991, pp. 681–688.
- <sup>6</sup>Korivi, V. M., Taylor III, A. C., Newman, P. A., Hou, G. W., and Jones, H. E., "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives," *Journal of Computational Physics*, Vol. 113, No. 2, 1994, pp. 336–346.
- <sup>7</sup>Elliott, J. and Peraire, J., "Constrained, Multipoint Shape Optimisation for Complex 3D Configurations," *Aeronautical Journal*, Vol. 102, No. 1017, 1998, pp. 365–376.
- <sup>8</sup>Kim, C. S., Kim, C., and Rho, O. H., "Sensitivity Analysis for the Navier–Stokes Equations with Two-Equation Turbulence Models," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 838–845.
- <sup>9</sup>Giles, M. B., Duta, M. C., and Müller, J.-D., "Adjoint Code Developments Using the Exact Discrete Approach," AIAA Paper 2001–2596, June 2001.
- <sup>10</sup>Nielsen, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163.
- <sup>11</sup>Nemec, M. and Zingg, D. W., "Newton–Krylov Algorithm for Aerodynamic Design Using the Navier–Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1146–1154.
- <sup>12</sup>Giles, M. B. and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, No. 3/4, 2000, pp. 393–415.
- <sup>13</sup>Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," AIAA Paper 2003–0272, Jan 2003.
- <sup>14</sup>Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," Lecture notes, von Karman Institute for Fluid Dynamics, Series: 1997-02, Brussels, Belgium, March 1997.
- <sup>15</sup>Melvin, R. G., Huffman, W. P., Young, D. P., Johnson, F. T., Hilmes, C. L., and Bieterman, M. B., "Recent Progress in Aerodynamic Design Optimization," *International Journal for Numerical Methods in Fluids*, Vol. 30, 1999, pp. 205–216.
- <sup>16</sup>Young, D. P., Melvin, R. G., Bieterman, M. B., Johnson, F. T., and Samant, S. S., "A Locally Refined Rectangular Grid Finite Element Method: Application to Computational Fluid Dynamics and Computational Physics," *Journal of Computational Physics*, Vol. 92, No. 1, 1991, pp. 1–66.
- <sup>17</sup>Dadone, A. and Grossman, B., "Efficient Fluid Dynamic Design Optimization Using Cartesian Grids," AIAA Paper 2003–3959, Orlando, FL, June 2003.
- <sup>18</sup>Nemec, M., Aftosmis, M. J., and Pulliam, T. H., "CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method," AIAA Paper 2004–0113, Reno, NV, Jan. 2004.
- <sup>19</sup>Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952–960.
- <sup>20</sup>Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA Paper 2000–0808, Reno, NV, Jan. 2000.
- <sup>21</sup>Haines, R. and Crawford, C., "Unified Geometry Access for Analysis and Design," Tech. rep., 12th International Meshing Roundtable, Santa Fe, NM, Sept. 2003.
- <sup>22</sup>Murman, S. M., Chan, W. M., Aftosmis, M. J., and Meakin, R. L., "An Interface for Specifying Rigid-Body Motions for CFD Applications," AIAA Paper 2003–1237, Reno, NV, Jan. 2003.
- <sup>23</sup>Dennis Jr., J. E. and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, N.J., 1983.
- <sup>24</sup>Moré, J. J. and Thuente, D. J., "Line Search Algorithms with Guaranteed Sufficient Decrease," *ACM Transactions on Mathematical Software*, Vol. 20, No. 3, 1994, pp. 286–307.
- <sup>25</sup>Haines, R. and Aftosmis, M. J., "On Generating High Quality "Water-tight" Triangulations Directly from CAD," Tech. rep., Meeting of the International Society for Grid Generation, (ISGG), Honolulu, HI, June 2002.

<sup>26</sup>Giles, M. B. and Pierce, N. A., "Analytic Adjoint Solutions for the Quasi-One-Dimensional Euler Equations," *Journal of Fluid Mechanics*, Vol. 426, 2001, pp. 327-345.

<sup>27</sup>Lu, J. and Darmofal, D. L., "Adaptive precision methodology for flow optimization via discretization and iteration error control," AIAA Paper 2004-1096, Jan 2004.

<sup>28</sup>Alonso, J. J., Martins, J. R. R. A., Reuther, J. J., Haimes, R., and Crawford, C., "High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model," AIAA Paper 2003-3429, Orlando, FL, June 2003.

<sup>29</sup>Jameson, A. and Kim, S., "Reduction of the Adjoint Gradient Formula in the Continuous Limit," AIAA Paper 2003-0040, Jan. 2003.

<sup>30</sup>Soto, O. and Löhner, R., "On the Computation of Flow Sensitivities from Boundary Integrals," AIAA Paper 2004-0112, Jan. 2004.

<sup>31</sup>Nielsen, E. J. and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411-1419.

<sup>32</sup>Huffman, W. P., Melvin, R. G., Young, D. P., Johnson, F. T., Bussoletti, J. E., Bieterman, M. B., and Hilmes, C. L., "Practical Design and Optimization in Computational Fluid Dynamics," AIAA Paper 93-3111, 1993.

<sup>33</sup>Aftosmis, M. J., Berger, M. J., and Murman, S. M., "Applications of Space-Filling-Curves to Cartesian Methods for CFD," AIAA Paper 2004-1232, Reno, NV, Jan. 2004.